REMARKS

Claims 1-11 are pending.

Pages 1, 4 and 5 of the specification have been amended in order to address an objection under 37 CFR 1.77(b) concerning section headings. Applicants note that a Summary of the Invention Section is not required by the MPEP.

Figure 3 is being amended in an attached sheet in response to its objection in the Office Action.

Claims 1 and 9 have been amended by incorporation of the features of claims 2 and 10, respectively, in order to clarify claims 1 and 9. Claims 2 and 10 have therefore been canceled. Claims 1 and 9 have additionally been amended in order to clarify the claims further in the light of the rejection under 35 USC § 101 (described in more detail below). No new matter is being added.

On page 2 of the Office Action, the Office Action objects to the specification under 37 CFR 1.77(b) for failing to contain section headings as required by 37 CFR 1.77(b). The specification is being amended to incorporate new headings. Applicants note that a Summary of the Invention Section is not required by the MPEP.

On page 3 of the Office Action, the Office Action objects to Figure 3 of the drawings for failing to comply with MPEP § 608.02(g) by not including the legend -- Prior Art --. The correction to Figure 3 is being made on an attached sheet.

On pages 3-5 of the Office Action, claims 1 to 11 are currently rejected under 35 USC § 101 as being unpatentable for being directed to non-statutory subject matter. Applicants are traversing this rejection.

The application presently contains two independent claims, namely claims 1 and 9.

Amended claim 1 is directed to a <u>decoder</u>. Furthermore, as amended, claim 1 now recites that the decoder is arranged to <u>receive an information bit</u> and to <u>use</u> a selected MAX* output value to <u>decode</u> the received information bit.

The Office Action cites MPEP § 2106.IV.B and re Schrader, 22F.3d 290, 295 (Fed. Cir. 1994), which is mentioned in MPEP § 2106.02, in support of this rejection. Lines 3-5 of numbered paragraph 3 of the Office Action states that:

"*each limitation in the decoder (apparatus) claim 1 is an abstract algorithm that can be carried by computer software program element and <u>is not tangibly</u> embodied.*"

It is assumed that this statement in combination with cited MPEP § 2106.IV.B implies that the Office Action is rejecting claim 1 on that grounds that it does not relate an enumerated statutory category. In this respect, 35 USC § 101 enumerates the statutory categories as any new or useful: process, machine, manufacture, or composition of matter.

Claim 1 is clearly directed to a <u>decoder</u>, which falls within the category of "machine" under 35 USC § 101. Because claim 1 is directed to a decoder, it not claiming "only a mathematical algorithm." Accordingly, it is respectfully submitted that claim 1 is directed to statutory subject matter.

Claims 2-8 depend from claim 1. By virtue of this dependence, claims 2-8 are also directed to statutory subject matter.

Turning to claim 9, page 4, line 21 – page 5, line 3 of the Office Action states:

"To qualify under section 101 statutory process, the claim should positively recite the other statutory class (the thing or product) to which the application is tied. See MPEP 2106.IV.B and In re Bilski 88 USPQ2d 1385 and In re Schrader 22F.3d 290, 295 (Fed. Cir. 1994)"

Amended claim 9 is now directed to <u>a method of decoding an information bit by a decoder</u> using a MAX* value. Amended claim 9 further recites the features of:

- receiving by the decoder, an information bit; and
- decoding by the decoder, the information bit using the selected value.

Amended claim 9 also now recites structural elements in the form of the decoder, a selector and a calculator. In the light of the introduction of these structural features into claim 9, it is submitted that the claim now positively recites "the other statutory class" to which the application is tied.

Accordingly, it is respectfully submitted that claim 9 is directed to statutory subject matter.

Claim 11 depends from claim 9. By virtue of this dependence, claim 11 is also directed to statutory subject matter.

On pages 3-4 of the Office Action, claims 1 and 13 are currently rejected under 35 USC § 103(a) as being unpatentable over US 6,757,703 (hereinafter referred to as "Sivan et al.") in view of US 6,922,711 (hereinafter referred to as "Kato et al."). Applicants are traversing this rejection.

As mentioned above, the application presently contains two independent claims, namely claims 1 and 9. Below, Applicants explain that the combination of Sivan et al. and Kato et al. does not teach all of the elements of claims 1 and 9.

Although Sivan et al. is the primary reference relied upon in the Office Action, Kato et al. will be described first as it describes the technical background particularly well.

Kato et al. relates to an "approximate calculator" for a non-linear function and an approximate calculator for a function log $(1 + e^{-x})$, and a Maximum A Posteriori (MAP) decoder employing the approximate calculator (col. 1, lines 15-18). As explained at col. 1, lines 26-28, a soft output decoder performs decoding using a MAP algorithm. By way of further background not described in Kato et al., the decoder is used to decode a received signal comprising turbo encoded information bits. The MAP algorithm is used in order to obtain decisions as to the value of a received encoded bit. An example of a MAP decoding process is shown in FIG. 9 of Kato et al. and comprises the calculation of probability density functions (pdfs) and probabilities α, β, and γ (col. 1, lines 49-53) corresponding to branches of a trellis used in such decoding processes. However, as explained at col. 1, lines 54-56, it is impractical to implement a MAP decoding process in hardware or software, because the number of bits required for representing intermediate results is large, resulting in larger circuitry to implement the decoding process.

In order to simplify computational requirements and hence reduce circuit size, a so-called "Log-BCJR" algorithm can be employed (col. 1, lines 64-67), the Log-BCJR algorithm being in the log domain, thereby replacing multiplication functions with summation functions.

As part of the process for calculating it is necessary to evaluate the expression: log $(e^a + e^b)$ in connection with the probabilities α, β, and γ. Furthermore, it is known that this expression is equivalent to max (a, b) + log $(1+ e^{-|a-b|})$ under certain conditions. As the equivalent expression comprises an exponential in the second term, it is relatively computationally complex to evaluate and so use of a look-up table is employed in order to obtain values for the second term (col. 2, lines 34-39). The hardware of such an implementation can be seen in FIG. 10 of Kato et al.

However, for an approximate value of log $(1+e^{-x})$ to be calculated using a look-up table, the look-up table needs to contain a relatively large number of sample points in order to maintain decoding precision (col. 2, lies 58-62). Consequently, as explained at col. 2, lines 64-65, the look-up table may be relatively large and may be computationally relatively complex.

In order to overcome the shortcomings of the look-up table approach, Kato et al. proposes a calculator that receives input data, x, and generates an approximate value, y, of the non-linear function using the input data, x. In the present case, the non-linear function is the exponential

term mentioned above. The calculator comprises a decoder 11, a shifter 12 and an adder 13. (col. 4, lines 35-39). In order to achieve the generation of the value, y, the non-linear function is linear-interpolated interval-by-interval as shown in FIG. 2 of Kato et al. (col. 4, lines 39-41) and straight lines conforming to the function Y=A.X + B are selected. The parameters A and B characterizing the straight lines selected are stored in a look-up table.

The decoder 11, receives the input data, x, and retrieves from a look-up table data that represents the gradient, A, of the linear function for the value of the input data, x. The decoder 11 also accesses the look-up table in order to obtain a value, B, for a Y-axis intercept (col. 4, lines 56-62). Referring to col. 4, lines 62-67, the shifter 12 then shifts the input data, x, by $|n|$ bits to the left or right, depending upon the sign and the value of the gradient, A, retrieved from the look-up table, thereby effectively performing the multiplication A.X, because A has a value of $\pm 2^n$, where n is an integer (col. 4, lines 41-42). The adder 13 then sums the calculated product A.x and the intercept data, B to yield the value, y required. It is pointed out that the shifting of the input data, x, <u>does not constitute the performance of modulo arithmetic</u> and even if it were considered execution of modulo arithmetic, Kato et al. <u>would still fail to disclose the performance of modulo arithmetic very specifically on the input data, x, and not in relation to any larger algebraic expression</u>.

A specific example in the context of the "formula" MAX (a, b) = log $(1+ e^{-|a-b|})$ is described from col. 5, line 33 onwards. In this respect, a calculator for calculating MAX (a, b) = log $(1+ e^{-|a-b|})$ is incorporated into a MAP decoder, the calculator being used to calculate the probabilities α and β using the Log-BCJR algorithm (col. 5, lines 33-37 and lines 41-43). The principle employed is the same as described above and so will not be described further for the sake of conciseness.

Turning to Sivan et al., this document relates to an apparatus and method for implementing a linearly approximated Log MAP algorithm (col. 1, lines 7-8).

Like Kato et al., Sivan et al. describes the existence decoders, the decoders being so-called Soft-In Soft-Out (SISO) decoders in the case of Sivan et al. Sivan et al. also explains that it is common for SISO decoders to employ a MAP or log MAP algorithm (col. 1, lines 34-42). As pointed out at col. 1, lines 44-49, another common decoding algorithm employed is the MAX log MAP algorithm, which involves an additional correction factor. Col. 2, lines 6-7 explains that in relation to calculating probabilities, the MAX* (a(n), b(n)) function is used and this function is equivalent to MAX (a(n), b(n)) + log $(1+e^{-|a(n)-b(n)|})$, the logarithmic term being obtained by calculating an approximation of the term (col. 2, lines 9-14). As stated at col. 2,

lines 16-24, linear approximation techniques perform poorly and are relatively time consuming to implement.

In contrast to Kato et al., Sivan et al. discloses a simple linear approximation of the exponential function $(1+e^{-|a(n)-b(n)|})$ that has a gradient of -0.5 (col. 2, lines 57-60 and col. 5, lines 5-6). According to a most relevant embodiment associated with FIG. 3, an apparatus 20 is provided that comprises first and second subtraction units 22, 23 and an adder 24 (col. 6, lines 10-11). The first and second subtraction units 22, 23 and the adder 24 receive values a(n) and b(n), associated with calculating probabilities, and a value DE. The adder 24 evaluates the expression (a(n)+b(n)+DE)/2 and provides a multiplexer 29 with the result. The first subtraction unit 22 generates the sign of (b(n)-a(n)-DE) and the second subtraction unit 23 generates the sign of (a(n)-b(n)-DE), the signs generated also being provided to the multiplexer 29, for controlling the output of the multiplexer 29 (col. 6, lines 11-17).

As explained at col. 5, lines 55-65, the apparatus 20 provides outputs according to equations [3], [4] and [5] depending upon the sign of (a(n)-b(n)-DE) or when $|a(n)-b(n)| <$DE. Hence, the MAX* function is implemented.

However, Applicants' repeat the point that Kato et al. fails to teach <u>to which</u> equation in Sivan et al. modulo arithmetic should be applied and additionally neither Sivan et al. not Kato et al. actually teach <u>how</u> the modulo arithmetic should be applied to any equation of Sivan et al., and in particular, equation [5] of Sivan et al.

Turning to claim 1, claim 1 recites a wireless communication device comprising:

- a calculator for calculating the modulo of a linear approximation of a MAX* function using $\left(a(n)\bmod F + \frac{((b(n)\bmod F - a(n)\bmod F)\bmod F + C)}{2}\right)\bmod F$; and
- a selector for selecting a MAX* output value from the group a(n)modF, b(n)modF, and the calculated modulo
- based upon a determination as to whether a predetermined threshold value for $|a(n)-b(n)|$ has been met, where a(n) is a first state metric, b(n) is a second state metric, C is the predetermined threshold value and F is a value greater than $|a(n)-b(n)|$; wherein
- the decoder is arranged to receive an information bit and

- to use the selected MAX* output value to decode the received information bit.

In the light of the above explanation, it is respectfully submitted that the combination of Sivan et al. and Kato et al. fails to teach a calculator for calculating a modulo of a linear approximation of a MAX* function using $\left(a(n)\bmod F + \frac{((b(n)\bmod F - a(n)\bmod F)\bmod F + C)}{2}\right)\bmod F$, as recited in claim 1.

In contrast, application of modulo arithmetic to yield the expression recited in claim 2 (now incorporated in claim 1) provides correct results. The teachings of Kato et al. do not provide the skilled person with instructions as to <u>which</u> expression disclosed in Sivan et al. to apply modular arithmetic (assuming, that is, that Kato et al. actually discloses the use of modulo arithmetic) and, moreover, neither Sivan et al. nor Kato et al. teach the skilled person as to <u>how</u> to apply modulo arithmetic correctly to equation [5] of Sivan et al. in order to obtain computationally correct results.

Page 6 of the Office Action alleges that all features of claim 1 are taught by Sivan et al. with the exception of the use of modulo arithmetic. The Office Action therefore relies upon the teachings of Kato et al. to allege the disclosure of modulo arithmetic in relation to a MAX* "formula". As mentioned above, Applicants' dispute that Kato et al. teaches use of modulo arithmetic. In any event, claim 1 has been amended for clarification purposes by incorporation of the feature of claim 2 currently on file.

Page 7, numbered paragraph 9, of the Office Action rejects the feature of claim 2, alleging that col. 5, equation [5] discloses the expression recited in claim 2. However, it is respectfully submitted that the Office Action overlooks the point made on page 6, lines 20-27 of the Applicants' specification, namely that direct application of modulo arithmetic upon the expression (a(n)+b(n)+C)/2, stated as (a(n)+b(n)+DE)/2 (equation [5]) in Sivan et al., <u>yields an incorrect result</u>. <u>This is the obvious approach</u>, not the approach recited in claim 1.

In view of the reasoning provided above, Applicant submits that Sivan et al. in view of Kato et al. does not render claim 1 obvious.

Claims 3-8 depend from claim 1. By virtue of this dependence, claims 3-8 are also not obvious.

Claim 9 is a method claim corresponding to the apparatus of claim 1. Consequently, the arguments set forth above in support of claim 1 apply equally to claim 9. As such, it is therefore respectfully submitted that the combination of Sivan et al. and Kato et al. fail to teach a

calculator calculating the modulo of a linear approximation of a MAX* function using: $\left(a(n) \bmod F + \frac{((b(n) \bmod F - a(n) \bmod F) \bmod F + C)}{2} \right) \bmod F$, as recited in claim 9.

In view of the reasoning provided above, Applicant submits that Sivan et al. and Kato et al. does not render claim 9 obvious.

Claim 11 depends from claim 9. By virtue of this dependence, claim 11 is also not obvious.

The case is believed to be in condition for allowance and notice to such effect is respectfully requested. If there is any issue that may be resolved, the Examiner is respectfully requested to telephone the undersigned.

Respectfully submitted,

SEND CORRESPONDENCE TO:

Freescale Semiconductor, Inc.
Law Department

Customer Number: 23125

By: /David G. Dolezal/
David G. Dolezal
Attorney of Record
Reg. No.: 41,711
Telephone: (512) 996-6839
Fax No.: (512) 996-6854